

Internship task

By code alpha

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Calculator</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="calculator">
    <input type="text" id="inputBox" placeholder="0" readonly>
    <div>
      <button class="operator">AC</button>
      <button class="operator">DEL</button>
      <button class="operator">%</button>
      <button class="operator">/</button>
    </div>

    <div>
      <button>7</button>
      <button>8</button>
      <button>9</button>
      <button class="operator">*</button>
    </div>

    <div>
      <button>4</button>
      <button>5</button>
      <button>6</button>
      <button class="operator">-</button>
    </div>

    <div>
      <button>1</button>
      <button>2</button>
      <button>3</button>
      <button class="operator">+</button>
    </div>

    <div>
      <button>00</button>
      <button>0</button>
      <button>.</button>
      <button class="equalBtn">=</button>
    </div>
  </div>
</body>
</html>
```

```
        </div>

    </div>
    <script src="script.js"></script>
</body>
</html>
```

```
// Select necessary elements
const inputBox = document.getElementById("inputBox");
const buttons = document.querySelectorAll("button");

let currentInput = ""; // To store the current input

// Function to update the input box
function updateInput(value) {
    inputBox.value = value || "0"; // Display "0" if the input is empty
}

function clearInput() {
    currentInput = ""; // Clear the input
    updateInput(currentInput);
}

function deleteLast() {
    currentInput = currentInput.slice(0, -1); // Remove the last character
    updateInput(currentInput);
}

// Event listener for button clicks
buttons.forEach(button => {
    button.addEventListener("click", () => {
        const value = button.innerText;

        // Check for operator buttons
        if (button.classList.contains("operator")) {
            if (value === "AC") {
                clearInput(); // Clear the input
            } else if (value === "DEL") {
                deleteLast(); // Remove the last character
            } else {
                // Add the operator to the input
                currentInput += value;
            }
        } else if (button.classList.contains("equalBtn")) {
```

```

        try {
            // Evaluate the input using eval (use with caution!)
            currentInput = eval(currentInput).toString();
            updateInput(currentInput);
        } catch (error) {
            // If there is an error in the input
            currentInput = "Error!";
            updateInput(currentInput);
            setTimeout(() => clearInput(), 1500); // Clear the input after
showing the error
        }
    } else {
        // Add the button value to the input
        currentInput += value;
    }

    // Update the display
    updateInput(currentInput);
});
});

```

```

body {
    font-family: Arial, sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
    background-color: #242323;
}

.calculator{
    border: 1px solid #717171;
    padding: 20px;
    border-radius: 16px;
    background: #242323;
    box-shadow: 0px 3px 15px rgba(113, 115, 119, 0.5);
}

```

```
input{
  width: 320px;
  border: none;
  padding: 24px;
  margin: 10px;
  background: transparent;
  box-shadow: 0px 3px 15px rgb(84, 84, 84, 0.1);
  font-size: 40px;
  text-align: right;
  cursor: pointer;
  color: #fff;
}

input::placeholder
{
  color: #fff;
}

button{
  border: none;
  width: 60px;
  height: 60px;
  margin: 10px;
  border-radius: 50%;
  background: #242323;
  color: #faf9f9;
  font-size: 20px;
  box-shadow: -8px -8px 15px rgba( 255, 255, 255, 0.1);
  cursor: pointer ;
}

.equalBtn{
  background-color: #fb7c14;
}

.operator{
  color: #6dee06;
}
```

